Q1 What is React?

Ans i) React is a JavaScript library for building user interfaces (Not Framework).

ii) It is used to build single-page applications.

iii) React allows us to create reusable UI components.

Frame Works V/s Library

(House bought) (House build by yourself).

Similarity

Both the library and the Framework are reusable pieces of code written to solve complicated problems.

Parameters	Library	Framework
Definition	Libraries provide developers with predefined classes and functions to make ~~their~~ work easier and boost the development process	Framework is like the foundation upon which developers build applications for specific platforms.
Inversion of Control	By using a library, you can control the flow of the application and call the library.	The framework controls the flow and calls your code for any custom behaviour.

Talk is cheap, show me the code

→ React element is an Object.

→ Naming convention } React element → all lowercase.

React component } First letter capital

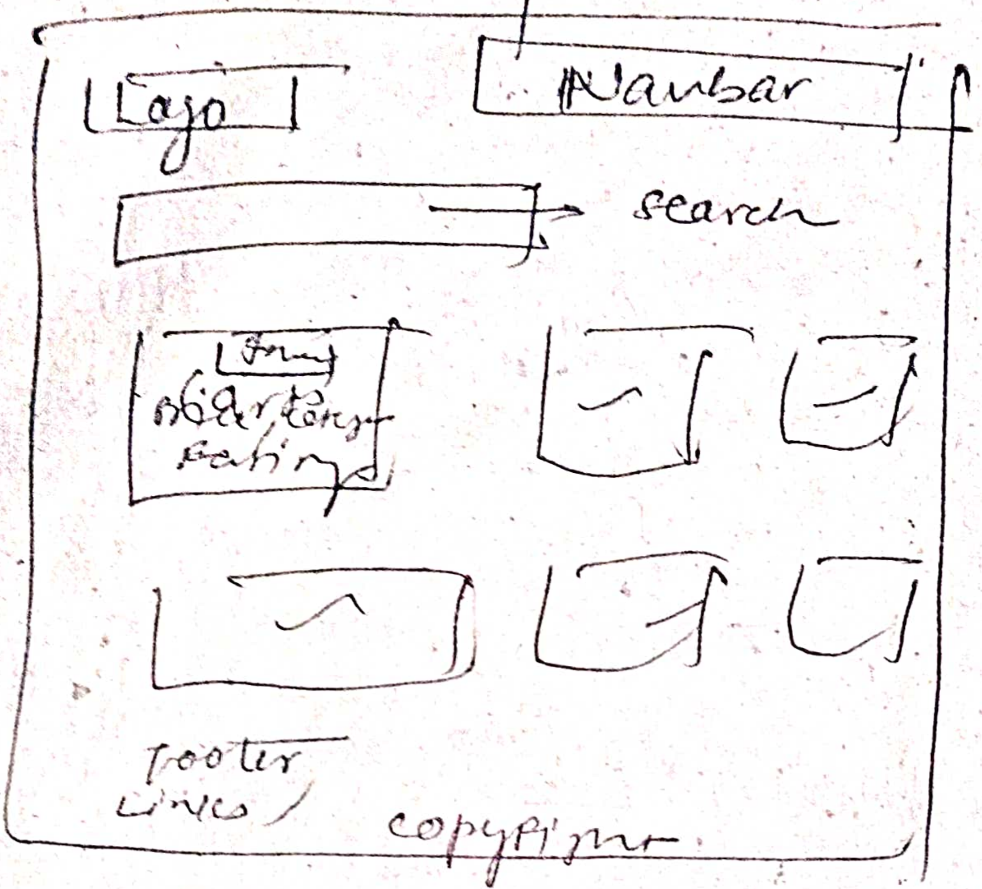
→ Any piece of javascript can be written inside JSX using only braces { }.

Q Is JSX mandatory?

Ans NO any tech stack is mandatory to use anything.

Never ever write code without planning.

Wireframing →



→ Building an Applayout component

const Applayout = () => {

return (

<

- Header
  - Logo
  - Navbar
  - Cart
- Body
  - Search bar
  - Restaurant List
    - Restaurant Card
      - Image
      - Name
      - Rating
      - cuisines
- Footer
  - Links
  - Copyright

→ JSX expression must have only on parent element.

Introducing React. Fragment

→ is a component exported by ~~export~~ react library way to use

```
<React. Fragment>
    [JSX]
</React. Fragment>
```

or

```
<>
[JSX]
</>
```

→ Its like an empty tag.

Use case ??

→ Inline styling in React

```
<div style = {styleObj}>
</div>
const styleObj = {
  background-color: "red",
}
```

Any piece of javascript can be written inside curly braces.

→ One can't give styling to React. fragment we have to wrap it inside div.

Q1 Is React Fragment can be nested ?

→ Building the Body Component

Q1 Ways to get data on UI ?
I) Hardcoded Data
II) By using APIs (integrating APIs).

→ Since we are making website where data changes at every instant we need to make it dynamic / Hardcoded data doesn't make sense.

→ Way to use Dynamic Data.

→ Join items in an Array.

→ Config Driven UI : The UI which is dynamic, i.e all UI will be driven by config sent by backend and config is driven by APIs and Backend.
(mention it using system design interviews)

→ Passing Dynamic Date (Different data) into Different Card,

Introducing Props

→ The things which are passed as attribute into the functional components.

Literally means, a way of passing data into the component.

// same as passing arguments into functions

→ You pass in arguments and receive parameter.

Qs can we pass multiple props at the same time?

Yes, we can ?

→ Object Destructing on the fly

props.restaurant = {restaurant}

→ Passing props using spread (...) operator.

→ map is the best way tot do functional programming:

- Difference b/w map and forEach?

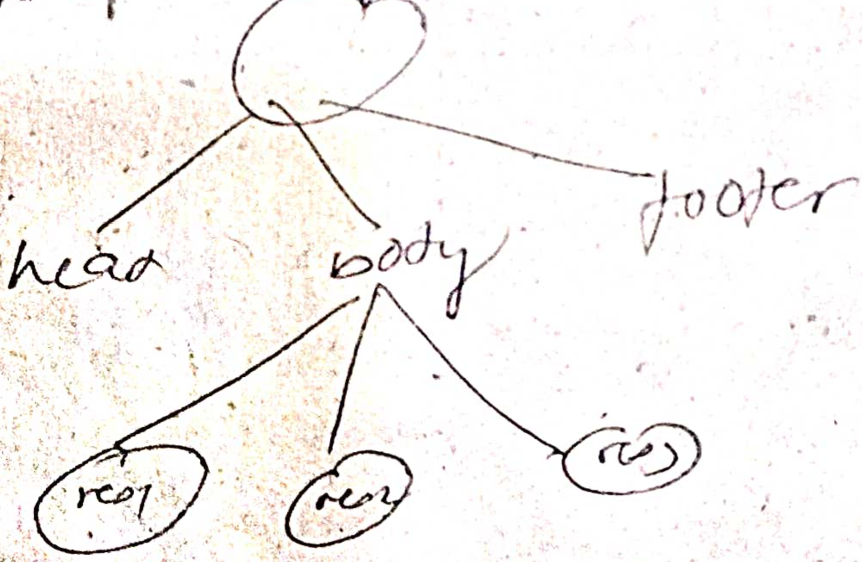
Each child in a list should have a unique key prop:

Introducing Virtual Dom

- It is a software engineering concept not of React.

Dom in ar core.

: Representation of



Q1 why do we need virtual DOM in React ?

Ans) It is used for the purpose of reconcillation (read the article on react.dev).

Reconcillation → happens using Diffing Algorithm

→ React uses its to differentenbate b/w ~~two~~ trees and on ~~that~~ it takes decision of ~~what~~ and what to change on UI and what not ~~to~~ change.

It only rerenders ~~those~~ parts which needs to be changed ~~this~~ not all → this makes React fast.

Multiple children with same ~~tag~~ name <div> then you need to pass a key

Because even if it knows something has been changed it will not be able to process without keys (only changed rerendering)

without keys → all elements rerendered.

with keys → only changed elements rerendered.

→ **React Fiber**

- new reconciliation engine for React (came in React 16)
- Responsible for all the DIFF.

reconciliation

The algorithm React uses to diff one tree with another to determine which parts need to be changed.

Never ever forget to write keys. (I only want to change what has been changed).

Analogy > If we want a room in our home, is it make sense to build the whole home? No na.

Q) Why it is bad to use index as keys in React? (React don't recommend index as keys)

no key <<<< index key << unique key (Best practice)